CATS
CAPITOL AREA TIMEX/SINCLAIR USERS GROUP

NEWSLETTER

$1.00 per copy
Please support your computer- JOIN CATS!

Vol 3, No. 4
July 1986

* * * CONTENTS * * *

From the Editor....

Welcome our New Officers!

New officers of the club were elected at last month's meeting. They were: Tom Bent, President; Hank Dickson, 1st VP; Harry Harrison, 2nd VP; and Ruth Fegley, Treasurer. I'll continue on as Editor, with Sarah Fisher and Bob Curnutt on the production end. We are still looking for a volunteer to supervise duplication of the club library tapes.

We'll need support from all of you to make the next year a success. If you have an idea for a meeting presentation, let Tom or Hank know. If you have an idea for an article, send it in! I want to especially thank Roald Schrack and H.E. Weppler for their consistent fund of new and interesting articles.

Prepare for August!

I'll be leaving town by July 18: articles for the August issue will have to reach me at least five days before that.

Last month's Meeting, and Next:

In addition to electing officers, we had an illuminating trio of talks by science fair contestants that had used Computers in their projects. Further details on page 2. I haven't talked with Tom to find out just what he's planning for next month, but I do know one thing - our meeting hall is AIR CONDITIONED! There'll be a cool space for you to come and join us and keep the spirit alive!

New Product Announcements:

If you have been going around in circles trying to hook up that unknown printer, LOGICAL SOLUTIONS of Gaithersburg (301) 977-1510 may be able to help build a cable to hook it up. They have a library of 1500 pin-outs for various devices.

CURRY COMPUTER has sent further details on their line of Pyramide software, to wit, QL PEINTRE. This program allows you to obtain graphics on the QL that rival those of the other 68000 machines. Also by Pyramide, VROOM, a graphic racetrack game, and OTHELLO. Also news of TECHNIQL, a CAD package for the QL, and ASSEMBLER WORKBENCH, a m/c assembler/ disassembler/ debugger. (602) 978-2902

TAS BAM User's Group has sent out news of a major upraade of Tom Wood's PRO FILE. With his permission, it has been adapted to the Rotronics Wafa-drive. The upgrade is available from Mr. George Fetherman, 5956 45th Ave. North, St. Petersburg FL 33709 for $11.00 incl s&h. (813) 546-4278 This is the upgrade only - full documantation is availabe from Tom Wood.

Mark Fisher

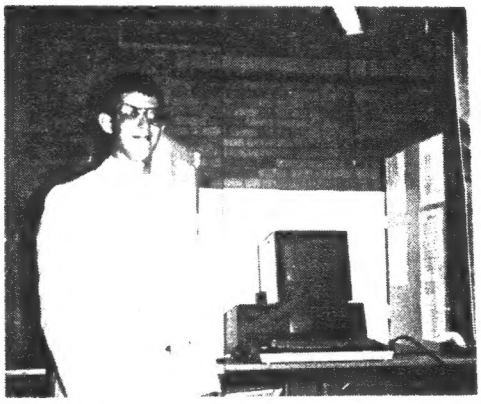
Science Fair Students Spark Fancy of CATS

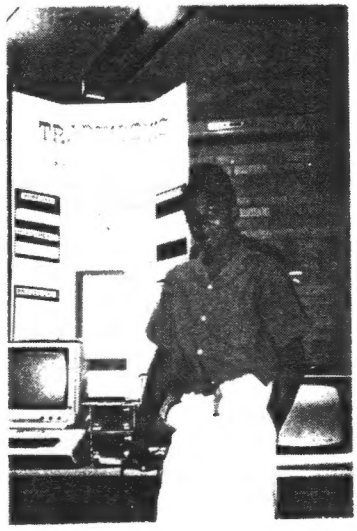
Featured at the June CATS meeting were three area science fair participants who described to the group how they successfully used small computers in carrying out their respective science projects.

These gifted young students brought their displays, set up all their equipmnent and gave extremely well-organized presentations, just as though they were addressing a large and friendly battery of judges.

The CATS members were fascinated.

We were also delighted to have as our guests the fathers of two of the exhibitors: Mr. AARON ENGEL and Mr. GEORGE L. WOOLEY, Sr. who, as it turns out, have been with their sons in the same Boy Scout troop for many years!





JOSHUA ENGEL, Eleanor Roosevelt HS, Greenbelt, MD
"An Easier PI"

Joshua became interested last summer in the mathematical abstraction known as "PI", and decided to make it the subject of his science fair project.

He described to us the refinement of pi from its earliest biblical reference down to the work of David Bailey who, at NASA's Ames Research Center last year, calculated pi to 29,360,128 places.

In searching through area libraries, including school, public, and U. of Maryland, he found volumes of material about pi, including some learned journals devoted to almost nothing else.

He also located 16 different "series" of pi, which "converged" with varying degrees of speed and ease. Thus some were "easier" than others, especially those he code-named "Pi-15" and "Pi-16".

He developed a computer program for his Osborne using the "C" language, and also implemented it using "C-Basic", which offered up to eleven decimal places for convergence. His Osborne was calculating some of these convergences while he was speaking.

Asked what standard for pi is used in space exploration, he replied that the classic 22/7ths is often good enough. During Voyager's probe of Uranus, using pi to 6 decimal places got the craft within 5 feet of its target. Using pi to 35 decimal places will, in calculating the circumference of the solar system, be acccurate to 1 one-millionth of the diameter of a proton, he pointed out.

When queried what he liked most about his subject, Joshua replied, "Pi is an irrational number, and I identity strongly with irrationality."

Joshua will be a junior next year.

GEORGE L. WOOLEY, Jr., Kenmoor MS, Landover, MD
"A Traptickovic Network"

As George explained, he happens to live in a house with a whole lot of small computers. So a natural thing to attempt for his science fair project was to try to make all of them communicate with each other.

He pulled together the following machines: (1) TRS-80, (2) Apple II, (3) Timex/Sinclair, (4) Commodore PET, and (5) VIC-20.

His purpose was to try to send and receive a simple message between all of them without using phone lines or modems.

He started digging into the the kinds of signals generated by the five computers in terms of timing, pitch and duration. He also examined these signals using an occilliscope.

After his empirical investigations, he concluded the two best candiates for computer-to-computer communication were the PET and the VIC-20. Of the others, the next closest was the TRS-80. He found that these three share the same general family of signals suitable for audio interfacing.

George proposed that for connecting a group of computers in the same school building, the most practical thing would be to have tape-oriented devices at each end to capture messages in a non-tended mode.

The audience was genuinely amused when George explained, for the second time, what a "Traptickovic" network was: the cryptographic use of the letters in all the machines he used.

Next year George will be in the 8th grade.

continued on p.3

Officers & Functionaries

President	Tom Bent
Vice-President	Hank Dickson
Vice-President	Harry Harrison
Treasurer	Ruth Fegley
N/L Editor	Mark Fisher
N/L Production	Sarah Fisher
	Bob Curnutt
Corresp Secy.	Mike Cohen

Contributors

Hank Dickson	H.L. Schaaf
Mark Fisher	H.E. Weppler
Roald Schrack	George White

Deadlines

Newsletter	Meeting Date
	July 12
July 16	August 9
August 29	September 13
September 26	October 11

DAVID KULP, W.T. Woodson HS, Fairfax, VA
"Acoustical Testing, An Engineering Experiment"

David's science fair project was inspired by the rebuilding of his school auditorium which created such bad acoustics that an acoustical engineer had to be called in to salvage it.

David is a percussionist in his school's music organizations, and is acutely aware of the absence or presence of good acoustics.

He posited it should be practical to put together a mathematical model which could predict what a finished room (or assembly hall) would produce in terms of acoustics.

He researched what constitutes acoustical acceptability, talked with four professionals in the field who deemed his undertaking unique, and set out.

Using his Sinclair 2068 and a 2040 printer, he created a singularly outstanding program. He found the key to the architechtual problems in this area were encapsulated in the factor called "reverberation time".

The program he developed uses environmental factors such as wall surface as well as the presence of chairs, carpeting, and people. After the 2068 operator has introduced these factors via non-hostile menus, bar charts can be produced depicting the resultant range of sound.

David also devised a LOGO-like language to produce a drawing of the room being analyzed which plots the movement and reverberation of the sound waves as though they were demonstrating fluid dynamics.

David says that after seven months, he is glad to set this project aside e and move on to other things. He doesn't know if his future portends engineering or computers, although he certainly has displayed skill at both.

David will be a senior next year.

At the conclusion of the presentations, the guests were presented small mementoes of their visit and were, by acclimation, voted into honorary membership in CATS. David Kulp, already a member of CATS, will have his current membership generously extended.

---C. D. Dickson

CRYPTOGRAM Solution:

IT MAY NOT BE VERY ETHICAL TO BLAME ALL
YOUR PROBLEMS ON THE COMPUTER, BUT
IT'S REALLY CONVENIENT

DUEL

By Mark Fisher

Prologue:

Long, long ago, in a city far away... A young man read of an intriguing algorithm for simulating motion on graph paper. This article was written by Martin Gardner in The Scientific American. At the time, it had been implemented on college mainframes in the form of a race track game, but the real interest was that anyone could implement it using nothing more than a pencil and graph paper.

As the young man was then going to Cal, and had nothing better to do, he extended the rules to simulate a battle between two ships in space. Unfortunately, the rules, while clear, were complex; and few people could master them well enough to see the elegance of the game. The young man soon learned that it was most interesting as a solitaire game, and gradually found other things to interest him....

In 1978, every Radio Shack in the country was displaying a computer that anyone could own. But what use was it? I couldn't think of anything that would justify $400.00....

In 1982, Timex put on its push for the TS 1000. Advertising leered out of all sorts of magazines, and real examples began to appear in local stores. It became obvious that what had seemed ridiculous claims for the ZX81 were true - and at under $100.00! I thought of that Spacewar game I'd fiddled with, and realized that if I programmed the rules into the computer, I wouldn't have to educate each person about addition of vectors before they could play the game. And so, mainly for that reason, I bought one.

Readers of this n/l may be aware that I've had a few other projects that developed after I got the 1000. In the meantime, I became comfortable with BASIC and machine code. Graphics, data handling, even animation might be needed to implement Spacewar. Slowly, as other projects waxed and waned, I built up an idea of what I wanted to do with the game.

The result is DUEL. It's not all I could wish for; I would like to see versions that allowed real-time movement, or that allowed two separate computers, linked through a MODEM, to control the "ships" in the game; but it's an engrossing introduction to the basic idea.

The Program

DUEL uses a number of the specialized abilities of the TS 2068, and a few tricks that speed up some of the BASIC.

1. Character arrays are used to store the coordinate sequences that keep track of the ships and missiles. This allows a single string slicing operation to shift the values to their new positions for each move, rather than a FOR-NEXT series of transfers.
2. DEF FN is used to quickly move numerical values out of the character array where they are stored.
3. ON ERR is used for two purposes: To allow the plotting of vectors that go off the screen without disrupting program flow, and later, to allow the 2068 to recover if a division by zero is encountered in calculating closest approaches.
4. STICK is decoded in an unusual way. As the bit values returned by STICK describe the position of the joystick, I POKE the value returned by STICK into address 16384 (which just happens to be the first byte in the display file), thus setting or resetting the first eight pixels of the d_file. I then use POINT to pick up the individual pixel values.
5. Mode OVER 1 is used. This allows me to erase PLOTted lines by re-PLOTting them.

Program flow

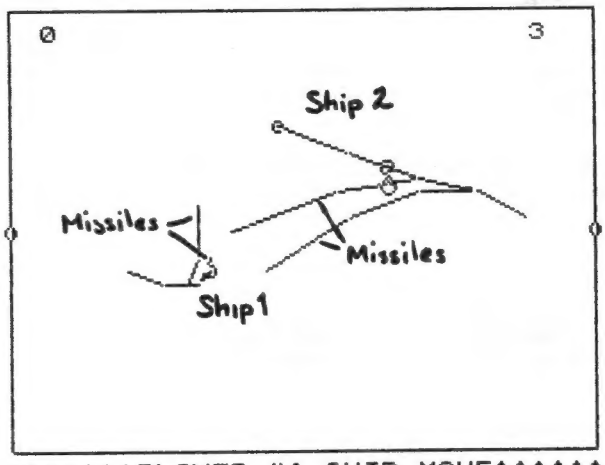
On RUN, the ON ERR and DEF FN flags are set, then the program jumps to the main title section (8000-8290). At this time data is initialized, and an option to view further instructions (8600-8830) is given. Following that, the operator has a choice of one or two joysticks. If (1) is pressed, both ship #1 and ship #2 are controlled by stick 1. If (2) is pressed, stick 1 controls ship #1 and stick 2 controls ship #2. Both the initialization and instructions are in high BASIC to speed up the operation of the main loop.

The program then cycles through the main loop (1000-1120), making excursions to the various subroutines as needed. The subroutine NEARMISS (3000-3090) was developed from formulas generated by Murray Barasch. In it the motion of two bodies is examined to see if they have reached a point of closest approach or not. I don't pretend to understand why it works.

The actual points of operation of the program are pretty well described in the REM statements. I needed them to keep track of what I was doing as I developed the program. Line numbers are regular because of the renumbering routine (9000-9120). The mc referred to in line 1100 and the SAVE-LOAD section (9610-) is the AERCO print driver. All functions of the program are in BASIC, and all references to the print driver can be left out.

If you get this going, let me know. I'd like to know how far my baby gets.

MF



Ship #2 was damaged two moves ago. As seen in the upper right corner, damage was 3. If damaged, a ship cannot maneuver, but it can still fire missiles. Damage is reduced one point per move. The dumbell shape records the positions of the missile and ship when the missile detonated. (Ship and missile tracks over four moves old are erased for clarity.)

continued on p.5

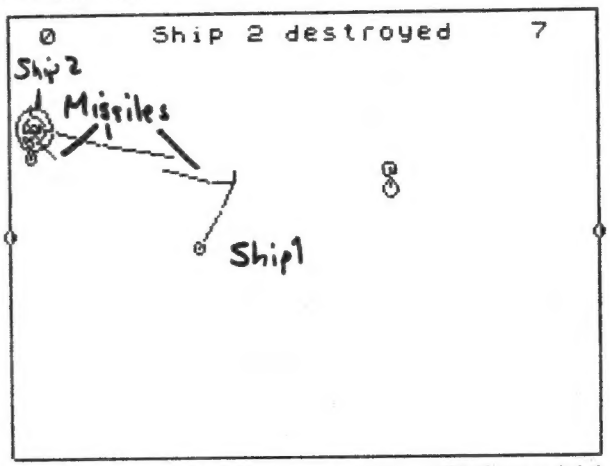
```
 10 REM ***   Duel   ****************************************
 20 REM  by Mark Fisher, (c) 1986    Math by Murray Barasch
 30 ON ERR  GO TO 9500
 40 DEF FN v(a,b,c)=CODE a$(a,b,c): REM Value
 50 DEF FN d(c)=FN v(a,b,c)-FN v(a,b,c+2)
 60 GO TO 8000
100 REM ** Get Coordinates ************************************
110 LET x=FN v(a,b,6): LET y=FN v(a,b,7): LET ox=FN v(a,b,8): L
ET oy=FN v(a,b,9)
120 LET dx=x-ox: LET dy=y-oy
130 RETURN
300 REM ******* stick reading ********************************
310 LET px=0: LET py=0: REM Pointer x,y
320 POKE 16384, STICK (1,r)
330 LET px=px+s*((POINT (4,175) AND px<2 AND px+x<=254)-(POINT
(5,175) AND px)-2 AND x-px)=1))
340 LET py=py+s*((POINT (7,175) AND py<2 AND py+y<=173)-(POINT
(6,175) AND py)-2 AND y-py)=2))
350 PLOT x,y: DRAW 8*px/s,8*py/s
360 LET ex= STICK (2,r)
370 PLOT x,y: DRAW 8*px/s,8*py/s
380 IF ex=0 THEN  GO TO 320
390 RETURN
1000 REM ****************** Main Loop ***************************
1010 FOR a=1 TO 2: IF q$="2" THEN  LET r=a
1020 INK a*2+3
1030 IF a$(a,1,3)=" COPY " THEN  GO SUB 2500: LET px=0: LET py=0
: GO TO 1090: REM ship gone but missiles active
1040 IF a$(o,1,3)=" COPY " AND a$(o,2,1)=" COPY " AND a$(o,3,1)=
" COPY " THEN  GO TO end: REM other ship gone and no missiles
1050 GO SUB 2000: REM get missile firing orders
1060 GO SUB 2500: REM move & update missiles
1070 INK a*2+2
1080 GO SUB 4000: REM get ship move
1090 GO SUB 4500: REM Move ship
1100 IF INKEY$="s" THEN  RANDOMIZE USR 64628: REM copy screen
1110 LET o=a: NEXT a
1120 GO TO 1000
2000 REM ************** Get Missile coords  *********************
2010 LET b=1: GO SUB 100: IF a$(a,1,2)=CHR$ 0 OR a$(a,1,3)=" COP
Y " OR a$(o,1,3)=" COPY " THEN  RETURN : REM find parent ship
2020 PRINT #1;AT 0,0;"))))))))PLAYER #";a;" MISSILE<<<<<<<Dist.
=";FN v(a,1,4);"x";FN v(a,1,5)," Missiles = ";FN v(a,1,2);" "
2030 LET s=1: GO SUB 300: REM s=step size
2040 PRINT #1;AT 0,0;,,
2050 IF NOT FN v(a,1,2) OR (px=0 AND py=0) THEN  RETURN : REM No
 missile fired
2060 LET a$(a,1,2)=CHR$ (FN v(a,1,2)-1)
2070 IF a$(a,2,1)=CHR$ 255 THEN  GO TO 2100: REM fire #2
2080 IF a$(a,3,1)<>CHR$ 255 THEN  LET b=3: LET tx=0: LET ty=0: P
LOT x,y: GO SUB abort: REM if two other, abort #3
2090 LET a$(a,3)=a$(a,2)
2100 LET a$(a,2)=CHR$ 8+CHR$ (px+3)+CHR$ (py+3)+a$(a,1,4 TO ): R
EM Fuel,x&y aim, and past movement
2110 RETURN
2500 REM ***************update missile movement*****************
2510 LET a=o: LET b=1: GO SUB 100: LET a=(NOT a-1)+1: LET sx=x:
LET sy=y: LET sox=ox: LET soy=oy
2520 FOR b=2 TO 3: GO SUB 100: IF a$(a,b,1)=" COPY " THEN  GO TO
 2620: REM If missile is inactive, skip update
2530 IF NOT FN v(a,b,1) THEN  LET a$(a,b,2 TO 3)=CHR$ 0+CHR$ 0:
REM if no fuel, no accel
2540 LET tx=(FN v(a,b,2)-3)*5+dx: LET ty=(FN v(a,b,3)-3)*5+dy: L
ET mx=tx+x: LET my=ty+y: REM Get new points..
2550 GO SUB nearmiss
2560 PLOT x,y: DRAW tx,ty
2570 IF t<=1 THEN  GO SUB check: GO SUB abort: GO TO 2620
2580 IF mx>253 OR mx<2 OR my>173 OR my<2 THEN  GO SUB abort: GO
TO 2620: REM and check that it is not off screen
2590 IF a$(a,b,12 TO 15)<>a$(a,1,12 TO 15) THEN  PLOT FN v(a,b,1
4),FN v(a,b,15): DRAW FN d(12),FN d(13): REM If track is differe
nt from ship, erase track
2600 LET a$(a,b,8 TO )=a$(a,b,6 TO ): REM move track data
2610 LET a$(a,b,4 TO 7)=CHR$ ABS dx+CHR$ ABS dy+CHR$ mx+CHR$ my:
 IF a$(a,b,1)>CHR$ 0 THEN  LET a$(a,b,1)=CHR$ (CODE a$(a,b,1)-1)
: REM Update fuel& nex points
2620 NEXT b
2630 LET b=1: RETURN
2800 REM ******* abort missile ********************************
2810 FOR c=14 TO 8 STEP -2: REM Erase aborted missile
2820   IF a$(a,b,c-2 TO c-1)<>a$(a,1,c-2 TO c-1) THEN  PLOT FN v
(a,b,c),FN v(a,b,c+1): DRAW FN d(c-2),FN d(c-1)
2830 NEXT c
2840 LET a$(a,b,1)=" COPY "
2850 PLOT x,y: DRAW tx,ty
2860 RETURN
3000 REM ******* nearmiss ***********************************
3010 LET mcos=(tx)/(SQR ((tx)*(tx)+(ty)*(ty)))
3020 LET msin=(ty)/(SQR ((ty)*(ty)+(tx)*(tx)))
3030 LET scos=(sx-sox)/(SQR ((sox-sx)*(sox-sx)+(soy-sy)*(soy-sy)
))
3040 LET ssin=(sy-soy)/(SQR ((soy-sy)*(soy-sy)+(sox-sx)*(sox-sx)
))
3050 LET mv=SQR (tx*tx+ty*ty)
3060 LET sv=SQR ((sox-sx)*(sox-sx)+(soy-sy)*(soy-sy))
3070 LET t=-((x-sox)*((mv*mcos)-(sv*scos))+(y-soy)*((mv*msin)-(s
v*ssin)))/((mv*mv+sv*sv)-(2*mv*sv*(mcos*scos+msin*ssin)))
3080 IF t<1 THEN  LET tx=tx*t: LET ty=ty*t
3090 RETURN
3200 REM ********** check miss *********************************
3210 LET mx=tx+x: LET my=ty+y
3220 LET sx=sox+t*sv*scos
3230 LET sy=soy+t*sv*ssin
3240 LET dx=sx-mx: REM Dist missile to opposing ship
3250 LET dy=sy-my: REM Dist missile to opposing ship
3260 IF ABS dx<8 AND ABS dy<8 THEN  DRAW dx,dy: CIRCLE sx,sy,3:
GO SUB boom
3270 RETURN
3500 REM ***** boom ***********************************************
3510 LET a$(o,1,3)=CHR$ (FN v(o,1,3)+2+(ABS dx<4)+(ABS dy<4))
3520 CIRCLE mx,my,3: PRINT #1;AT 0,0;"*()<>)=()*/ SHIP #";o;" HI
T ,/-^?`#$Damage =";FN v(o,1,3),"Dist. ";INT dx;" x ";INT dy;"
": PAUSE 200
3530 LET d(o)=d(o)+FN v(o,1,3): PRINT  OVER 0;AT 1,2;d(1);TAB 28
;d(2): IF FN v(o,1,3)<4 THEN  RETURN
3540 FOR i=2 TO 10 STEP 3: CIRCLE mx,my,i: NEXT i
3550 PRINT  OVER 0;AT 1,7;" Ship ";o;" destroyed ": IF a$(o,1,3)
<>" COPY " THEN  LET k(a)=k(a)+1
```

continued on p.6

```
3560 LET a$(o,1,3)=" COPY "
3570 IF a$(o,2,1)=" COPY " AND a$(o,3,1)=" COPY " THEN  GO TO en
d: REM if no active missiles, stop.
3580 RETURN
4000 REM ****** Ship Move ************************************
4010 GO SUB 100: REM find parent ship
4020 PRINT #1;AT 0,0;"^^^^^^^^PLAYER #";a;" SHIP MOVE^^^^^^^"; FLA
SH FN v(a,1,3))1;"Damage =";FN v(a,1,3); FLASH FN v(a,1,1)<20;"
Fuel =";FN v(a,1,1);TAB 21;"Mssl = ";FN v(a,1,2)
4030 IF a$(a,1,1)=CHR$ 0 OR a$(a,1,3))CHR$ 1 THEN  PAUSE 80: LET
 px=0: LET py=0: RETURN : REM if damage or no fuel, no accel
4040 LET s=2: GO SUB 300: LET px=px/2: LET py=py/2: REM coarser
step than missile
4050 RETURN
4500 REM **********update ship movement **************************
4510 PLOT FN v(a,1,14),FN v(a,1,15): DRAW FN d(12),FN d(13): CIR
CLE FN v(a,1,6),FN v(a,1,7),2: REM erase track
4520 IF a$(a,1,3)=" COPY " THEN  LET a$(a,1,8 TO )=a$(a,1,6 TO )
: GO TO 4620
4530 LET mx=px*5+dx+FN v(a,1,6): LET my=py*5+dy+FN v(a,1,7): REM
 Get new points..
4540 LET a$(a,b,1)=CHR$ (CODE a$(a,b,1)-5*(ABS px) AND CODE a$(a
,b,1))
4550 LET a$(a,b,1)=CHR$ (CODE a$(a,b,1)-5*(ABS py) AND CODE a$(a
,b,1)): REM reduce fuel if burned
4560 IF FN v(a,b,3))0 THEN  LET a$(a,b,3)=CHR$ (CODE a$(a,b,3)-1
): REM reduce damage
4570 IF mx)253 OR mx<3 OR my)173 OR my<3 THEN  LET a$(a,1,3)=CHR
$ 4: LET o=a: GO TO boom: REM and check that it is not off scree
n
4580 LET dx=ABS (FN v(a,1,6)-FN v(o,1,6)): REM Dist to opposing
ship
4590 LET dy=ABS (FN v(a,1,7)-FN v(o,1,7)): REM Dist to opposing
ship
4600 LET a$(a,1,4 TO )=CHR$ dx+CHR$ dy+CHR$ mx+CHR$ my+a$(a,1,6
TO )
4610 PLOT FN v(a,1,8),FN v(a,1,9): DRAW FN d(6),FN d(7): CIRCLE
FN v(a,1,6),FN v(a,1,7),2
4620 RETURN
8000 REM ******************** Setup ****************************
8010 OVER 1: BORDER 0: PAPER 0: INK 7: CLS
8020 GO SUB 8500
8030 DIM a$(2,3,15): DIM k(2)
8040 DIM d(2): LET x=0: LET y=0: LET b=1: REM x,y=screen coord,b
=ship or missile select flag
8050 LET dx=0: LET dy=0: LET o=2: REM x&y movement from last pos
; o=other player
8060 RESTORE 8100: FOR a=1 TO 2: FOR c=1 TO 15
8070 READ x
8080 LET a$(a,1,c)=CHR$ x
8090 NEXT c: NEXT a
8100 DATA 250,10,0,255,255,3,87,3,87,3,87,3,87,3,87: REM ship #1
8110 DATA 250,10,0,255,255,252,87,252,87,252,87,252,87,252,87: R
EM ship #2
8120 FOR b=2 TO 3
8130 LET a$(1,b)=CHR$ 255+a$(1,1,2 TO ): LET a$(2,b)=CHR$ 255+a$
(2,1,2 TO )
8140 NEXT b
8150 LET abort=2800
8160 LET nearmiss=3000
8170 LET check=3200
8180 LET boom=3500
8190 LET end=9560
8200 PRINT #1;AT 0,0;"(I)nstructions, (1)-(2) Sticks";,,,: PAUSE
 0: LET q$=INKEY$: LET r=1: CLS
8210 IF NOT (q$="1" OR q$="2") THEN  GO TO 8600
8220 CIRCLE 252,87,1: CIRCLE 3,87,1: PLOT 3,3: DRAW 249,0: DRAW
0,169: DRAW -249,0: DRAW 0,-169
8230 PRINT AT 1,2;d(1);TAB 28;d(2): LET b=1: FOR i=1 TO 2
8240 FOR a=1 TO 2
8250 GO SUB 100: READ px: READ py
8260 DATA 1,0,-1,0,1,0,-1,0,1,-1,-1,1,1,-1,-1,1
8270 GO SUB 4500
8280 NEXT a: NEXT i
8290 GO TO 1000
8500 REM ******* titles ***************************************
8510 PRINT AT 2,10;"************                    *   Duel   *
             ************"
8520 PRINT AT 5,15;"by","          Mark  Fisher","
 1985"''" You are fighting a space duel.  Each ship has 10 missi
les and   equal fuel."
8530 PRINT " Missiles have proximity fuses-  they will explode i
f they pass  within 8 units."
8540 PRINT " If damaged, you cannot change   course, but you can
 still fire  missiles."
8550 PRINT " If you go out of bounds, you    will be destroyed."
8560 RETURN
8600 REM ********** Instructions ********************************
8610 CLS : PRINT "         Joysticks"''"Joysticks control direc
tion of  thrust of ships and missiles."
8620 PRINT '"Actual direction is the functionof total accelerati
on applied tothe vehicle."
8630 PRINT '"The missiles are aimed through  16 points:"
8640 DATA 2,0,2,-1,2,-2,1,-2,0,-2,-1,-2,-2,-2,-2,-1,-2,0,-2,1,-2
,2,-1,2,0,2,1,2,2,2,2,1
8650 RESTORE 8640: FOR i=1 TO 16: PAUSE 5: PLOT 40,60: READ x: R
EAD y: DRAW 10*x,10*y: NEXT i
8660 PRINT AT 18,0;"And ships are aimed through 8   points:"
8670 DATA 1,0,1,-1,0,-1,-1,-1,-1,0,-1,1,0,1,1,1
8680 RESTORE 8670: FOR i=1 TO 8: PAUSE 10: PLOT 210,60: READ x:
READ y: DRAW 10*x,10*y: NEXT i
8690 PRINT "Press FIRE to enter desired vec-tor.";
8700 PAUSE 100: PRINT #1;AT 0,0;"Try it now, using STICK 1. Pres
s FIRE for more:"
8710 CIRCLE 130,60,3: LET r=1: LET s=1: LET x=130: LET y=60: GO
SUB 300
8720 CLS : PRINT '"Aiming missiles:"''"1) If aim vector is not p
lotted,   a missile will not be fired.    To skip missile firing
, press   FIRE without moving stick."
8730 PRINT '"2) A ship may only maintain 2      missiles: If a t
hird is fired   the oldest missile will be      aborted."
8740 PRINT '"3) Initial missile direction is    determined by th
e ship's        motion, added to the missile    aim vector."
8750 PRINT '"4) Missiles will be aborted when   they no longer a
pproach         target."
8760 PRINT #1;"(P) to play, (ENTER) to continue": PAUSE 0: LET q
$=INKEY$: IF q$="p" THEN  GO TO 8040
8770 CLS : PRINT "Ships"''"Ship speed is built up by suc-  cessi
```

continued on p.7

```
ve foreward accelerations. If you accelerate five times    forew
ard, "
8780 DATA 150,10,0,255,255,30,87,30,87,30,87,30,87,30,87: REM sh
ip #1
8790 RESTORE 8780: FOR c=1 TO 15: READ x: LET a$(1,1,c)=CHR$ x:
NEXT c
8800 LET a=1: LET o=2: LET b=1: FOR i=1 TO 5: GO SUB 100: LET px
=1: LET py=0: GO SUB 4500: NEXT i
8810 INPUT "Press ENTER";q$: PRINT '"- it will take five moves t
o    return to a stop."
8820 LET px=-1: FOR i=1 TO 5: GO SUB 100: GO SUB 4500: NEXT i
8830 GO TO 8040
9000 REM ********** Renumber and Save **************************
9010 INPUT "Start ";start
9020 INPUT "End ";end
9030 IF end)=9000 THEN  LET end=9000
9040 INPUT "Start renumber using ";n
9050 LET x=PEEK 23635+256*PEEK 23636
9060 IF PEEK x*256+PEEK (x+1)<start THEN  GO TO 9110
9070 IF PEEK x*256+PEEK (x+1))=end THEN  STOP
9080 POKE x,INT (n/256)
9090 POKE x+1,n-PEEK x*256
9100 LET n=n+10
9110 LET x=x+4+PEEK (x+2)+PEEK (x+3)
9120 GO TO 9060
9500 REM ************ error handling ****************************
9510 ON ERR  RESET
9520 IF PEEK 23739=21 OR PEEK 23739=9 THEN  GO TO 9560: REM repl
ay?
9530 IF PEEK 23739=6 THEN  LET t=1
9540 ON ERR  GO TO 9500
9550 GO TO PEEK 23736+256*PEEK 23737+1
9560 REM ***** replay ***************************************
9570 PRINT AT 3,8;"Score is ";k(1);" to ";k(2);TAB 8;"Play again
? (y/n)"
9580 IF INKEY$="y" THEN  GO TO 8040
9590 IF INKEY$="n" THEN  STOP
9600 GO TO 9580
9610 REM ***** start up *************************************
9800 CLEAR 64255: LET p=64261: POKE 26704,INT (p/256): POKE 2670
3,p-(INT (p/256)*256): LOAD ""CODE
9810 RUN
9900 CLEAR : SAVE "duel" LINE 9800: SAVE "prcode"CODE 64256,1111
9910 PRINT "rewind and verify:"'': VERIFY "duel": VERIFY "prcode
"CODE
9999 INK 7: LIST 2500
```



↑↑↑↑↑↑↑PLAYER #1 SHIP MOVE↑↑↑↑↑↑↑
Damage =0 Fuel =135 Mssl = 2

As shown, ship #2 has now been destroyed. Ship #1 must still evade the two missiles fired by #2, however.

CELESTIAL MECHANICS ON THE 2068

The Halley's comet code inspired me to examine celestial mechanics further. The Halley code was based on the known elliptical path of the comet. Newton showed that this elliptical path was caused by the inverse square law of gravitational attraction. That is, the gravitational force F between any two bodies is of the form:

F= g*m1*m2/(r*r) where the m's are the masses of the two bodies, g is the gravitational constant and r is the distance between the bodies. In this code one of the masses represents the sun, ms. The other mass can represent a satellite of the sun, a planet, a comet, or an interloper from outside our solar system. If the satellite is "captured" by the sun and cannot escape, the orbit is an ellipse; if the satellite can escape, then the orbit is a hyperbola. Other force laws will result in different orbits.

In this code you choose the starting position and velocity, and the satellite will then describe an orbit determined by the inverse square law force. When the satellite is nearest the sun the total energy is calculated. A "+" or "-" is written in the upper left-hand corner of the screen to indicate the energy of the sattelite. If "+" then the satellite will escape; if "-" it will be captured. I have written the code so that the path is only within the screen boundaries but this could be altered. The starting value of x has been made 127 but this also can be changed to suit your preference.

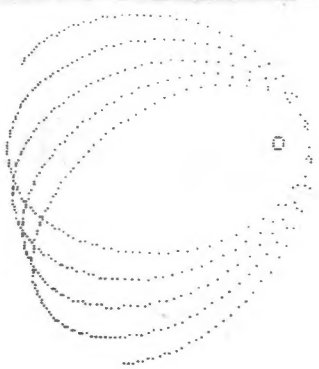
Note that the space between the succesive orbit positions is further apart as the satellite nears the sun showing that it travels with higher velocity. The code thus also illustrates Kepler's laws of planetary motion. Orbit calculations are very sensitive to accumulated error, so the precision of the 2068 will restrict the accuracy of the calculations. A nice repeating elliptical orbit is obtained for y=10,vx=1 but reducing vx to 0.6 will show precessing orbits due to the introduction of errors. Keeping y=10 and increasing vx to values greater than 1.42 will result in "escape" hyperbolic orbits with positive energy.

One can change line 125 to represent different force laws such as 1/r and 1/r^3. The accompanying table gives the starting parameters for some sample cases.

Roald A. Schrack

See also "Double Sun" Sept. '84 CATS Newsletter

```
 10 REM * Celestial Mechanics *
 20 LET dt=3
 30 LET m=10: LET ms=100
 40 INPUT "Give Y, 0 to 175 ";y
 50 INPUT "Give Velocity ";vx
 52 LET t=0
 60 CLS
 62 LET rl=255
 64 LET xl=0
 66 LET yl=0
 70 CIRCLE 190,87,2
 80 LET ax=0: LET ay=0
 90 LET x=127: LET vy=0
100 LET r2=(x-190)*(x-190)+(y-87)*(y-87)
120 LET r=SQR r2
125 LET f=ms*m/r2
130 LET fx=-f*(x-190)/r
140 LET fy=-f*(y-87)/r
150 LET ax=fx/m: LET ay=fy/m
160 LET vx=vx+ax*dt
170 LET vy=vy+ay*dt
172 LET x=x+vx*dt
174 LET y=y+vy*dt
175 IF t=1 THEN GO TO 180
176 GO SUB 300
178 LET rl=r: LET xl=x: LET yl=y
180 IF x<0 OR x>255 THEN GO TO 40
190 IF y<0 OR y>175 THEN GO TO 40
200 PLOT x,y
210 GO TO 100
300 IF r<rl THEN RETURN
310 LET dd=(x-xl)*(x-xl)+(y-yl)*(y-yl)
320 LET vt2=dd/(dt*dt)
330 IF vt2>2*100/r THEN GO TO 360
340 PRINT "-": LET t=1: RETURN
360 PRINT "+": LET t=1: RETURN
```



SAMPLE ORBIT PARAMETERS

Y	X	VX	DT	MS	F	note
10	127	1	3	100	1/r^2	a
10	127	.6	3	100	1/r^2	b
10	10	1.1	3	100	1/r^2	c
25	190	1.3	3	100	1/r^2	d
50	127	1.0	0.5	10	1/r	e
30	127	0.26	3	100	1/r^3	f
75	127	1.0	3	100	1/r^3	g

Notes: for the first two cases use the code as shown. The values of X,DT,MS,and the force law must be changed in the code. a. is a typical ellipse b. is a precessing ellipse c. is a hyperbola d. is a circle e. petal shaped orbits f. double loop g. hair pin These samples are just what I found in a few minutes. Vary the parameters and see what you find.

VU-CALC 2068
ROOM ! FOR MORE FORMULA !

As it comes from TIMEX(psion) you can enter 100 formulas, then "No more room for formulas" will keep you from going further.

Only 3 POKES are needed to get 128 formulas.

From a "cold" program with an empty spreadsheet, use #0 to get out of the spreadsheet, then 1 to get to BASIC.

Now make the following POKES:

```
POKE 30161,0
POKE 30162,1
POKE 33667,128
```

GOTO 9000 takes you into a new VU-CALC spreadsheet with room for 128 formulas.

To salvage a spreadsheet with data and formula is possible. Here's how I did it:

Quit to basic as above, and then entered everything in the immediate mode so as not to mess with any line numbers.

First find where the data ends by a PRINT PEEK bfre + 256*PEEK(bfre+1) Make a note of this. I'll refer to it as nnnnn, as it is a 5 digit number.

Now move all the formula up by 56 bytes (2 for each new formula) as follows

```
FOR i=nnnnn TO 54826 STEP -1:
POKE(i+56),PEEK i: NEXT i
```

The formula pointers also have to be increased by 56 like so:

```
FOR i=54626 TO 54825 STEP 2:
LET a=56+PEEK i + 256*PEEK
(i+1): POKE (i+1),INT(a/256):
POKE i,a-256*PEEK(i+1):NEXT i
```

Now add 56 to the bfre pointer:

```
LET a=56+PEEK bfre+256*PEEK
(bfre+1): POKE(bfre+1), INT
(a/256): POKE bfre,a-256*
PEEK(bfre+1)
```

Now make the same 3 POKES that were used for the blank VU-CALC spreadsheet.

GOTO 9000 should now let you enter 28 more formula and work with your existing data.

There are a couple of things we can do now to "enhance" the program by having it inform us about our formula room when we use the #0, and some other tidy messages. Here's a listing for lines I've changed to suit me.

```
1000 GO SUB VAL "1100": LOAD a$C
ODE : CLS : GO TO USR e2
1100 GO SUB VAL "1200": PRINT '
'"ENTER data file name": INPUT
a$: PRINT a$: RETURN
3000 GO SUB 1200: PRINT AT 4,0;P
EEK 33667;" formula-";PEEK 34566
;" in use=";(PEEK 33667)-PEEK 34
566;" open";AT 9,2;"ENTER 1 : E
XIT PROGRAM"'TAB 9;"2 : CLEAR W
ORKSHEET"'TAB 9;"3 : RETURN TO
VU-CALC": INPUT "OPTION? ";a
3200 CLEAR : DIM b$(100): DIM c$
(20): GO SUB VAL "1200"
9000 LET a=PEEK xrow: LET b=INT
(a/26): LET a=a-b*26: LET d$=CHR
$ (b+64)+CHR$ (a+65): IF b=0 THE
N LET d$=d$(2 TO 2)
9100 PRINT '''"ERROR was at",D$
;PEEK XCOL+VAL "1"
9200 INPUT "Press ENTER to conti
nue";b$: GO TO USR e2
```

To save it, I used:SAVE "vu128 calc" LINE 10 and SAVE "vucal c" CODE 29328,25555

I was able to figure this out by using HOT Z on the NV Memory board built at the Saturday CATS hardware sessions. That's a powerful combination !!

H. L. Schaaf

VARIABLES IN INPUT PROMPTS

This is not a new discovery but, because it is not mentioned in the 2068 User Manual, I feel that it is not used as widely as it might be.

Variables may be included in an Input prompt by enclosing the prompt in parentheses. That is, with INPUT (X);B the prompt will be the value of variable X and with INPUT (Y$);B it will be the string value of variable Y$.

It is feasible to combine numeric and string variables, along with other characters. For example, in an accounting program the prompt might be:

```
INPUT ("#";N;"  "+A$(N);TAB 22;
"$");B(N)
```

Then, when N=4 and when N=5 the prompts would appear as:

```
#4 INTEREST REC.      $
#5 DIVIDENDS REC.     $
```

The amounts entered would be stored in B(4) and B(5) respectively.

H.E.Weppler

CRYPTOGRAM

YM NOT DEM PS ASHT SMZYLOR ME

PRONS ORR TECH XHEPRSNU ED

MZS LENXCMSH, PCM YM'U HSORRT

LEDASDYSDM.

MARYLAND HAMFEST

AND COMPUTERFEST

SPONSORED BY THE

BALTIMORE RADIO AMATEUR TELEVISION SOCIETY

SUNDAY, JULY 27, 1986

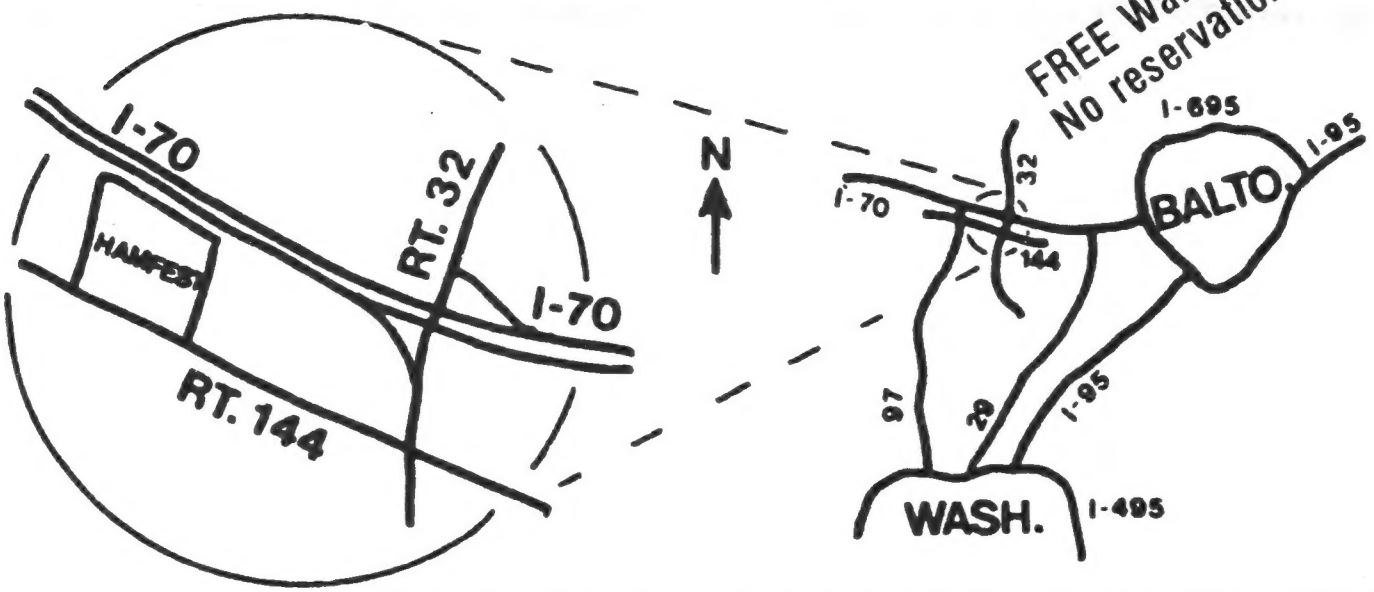
OPEN FOR SET-UP, SUNDAY AT 6 A.M.
DEALER SET-UP BEGINS SATURDAY AT 2 P.M.

AT THE

HOWARD COUNTY FAIRGROUNDS

15 miles WEST of BALTIMORE
just off I-70 at RT. 32 on RT. 144
ACCESSIBLE TO THE HANDICAPPED

FREE Walk-in VE Exams
No reservations Needed!



TOP PRIZES GIANT FLEA MARKET INDOOR/OUTDOOR EXHIBIT AREAS
GOOD FOOD REFRESHMENTS COMPUTERS RAIN OR SHINE

TALK-IN: 146.16/76
147.63/03
146.52

TICKETS: $4
TAILGATING
TABLES AVAILABLE

FOR INFO & RESERVATIONS,
WRITE: BRATS
P.O. BOX 5915
BALTIMORE, MD 21208

KIDS UNDER 12 FREE!

Capitol Area Timex/Sinclair Users' Group
P.O.Box 725
Bladensburg, MD 20710

Name _____

Address _____

_____ ZIP _____

Phone Home _____ Office _____

Memberships - $15.00 (family/individual);make checks payable to C.A.T.S.

If family membership, please list family members participating:

_____

Occupation _____

Ham Radio call sign _____

Equipment

ZX 80 _____ RAM size _____

MA 80 _____ full keyboard _____

ZX 81 _____ Printer _____

TS 1000 _____ type _____

TS 2000 _____ other interface _____

Special interest use for computer: ie, games, ham radio interface, business, other, etc. _____

_____

Languages: Basic _____ Other _____

Machine _____ _____

No. of years computer experience _____

What committees would you like to serve on? _____

_____

Comments: Where did you hear of C.A.T.S?

_____

- -

Do not write below:

Dt. Pd. _____ Amt. _____ Membership No. _____

Ca. _____ Ck. _____

The mailing address of the Capitol Area Timex/Sinclair User's Group is:

Capitol Area Timex/Sinclair User's Group
P.O. Box 725
Bladensburg, MD 20710

CATS is a non-profit special interest organization dedicated to serving the interests of those who own, use, or are interested in learning more about the Timex/Sinclair family of personal computers.

The official contact person for CATS is JULES GESANG:

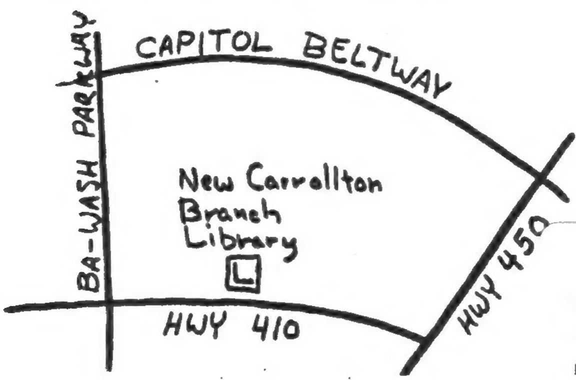
301#922-0767

Meetings are held on the second Saturday of each month at 2 P.M. in the large meeting room of the New Carrollton Branch Public Library.

Ham Radio Network Information

OZX Net...Wednesdays, 9p.m. local time; 14.345 MHz NV4F NCS
Eastern Regional Sinclair Net...Sundays, 1600 Z; 7.245 MHz
KO2F NCS

CATS Newsletter
P.O. Box 725
Bladensburg MD 20710



QUINTERO,MANUEL A. 301
4375 GREENBERRY LANE
ANNANDALE VA 22003

COME TO OUR MEETING!
The next meeting of C.A.T.S. will be held on:

Saturday, July 12, 1986

At: New Carrollton Public Library
7414 Riverdale Road (Hwy 410), New Carrollton, MD

IF YOU ARE NOT A MEMBER OF CATS, THIS IS THE ONLY ISSUE YOU WILL RECIEVE

Dues = $16.00 per year, per family.

First Class